sDotson.com

Autodesk Inventor Tutorials

by Sean Dotson
www.sdotson.com
sean@sdotson.com

How to Setup Your Project File for iParts

Latest Revision: 10/7/02

In this tutorial we will discuss a few ways to correctly set up your project file to use iParts. There are many different ways to set them up and I will discuss the two most common ones in this document.

First some basics about iParts. I think the help in Inventor describes them fairly well:

"In family-of-parts publishing, you create multiple instances of a part, varying its parameters and properties as necessary in each instance, so that the same design can be used repeatedly. To manage the multiple instances, and to enable you to easily switch to a different instance, part parameters and properties are stored in a spreadsheet. The collection of parts in the spreadsheet is known as a family-of-parts."[1]

Basically iParts are regular parts that can change size and shape, even suppressing and computing features based on the data contained in an embedded spreadsheet. The basic iPart, know as the "factory" or "parent" stores this spreadsheet internally. When you choose to place this factory part you select from a set of predefined values for size, shape and configuration. This then creates a "child" part. (I will be referring to this parent-child relationship through the tutorial). Both the locations of the parent and child parts need to be defined in your project path file.

There are two basic schools of thought as to where the child parts will be stored. I have named them the global and project specific methods.

In the global method the child parts are stored in a central location available to every project and to every user on the network. This type of structure is useful when building standard equipment that uses the same iPart children again and again. It also has the advantage that once it has been used for a while many of the common children have been created and the placement process is slightly faster.

In the project specific method, the child parts are stored in a project path folder. This folder changes for each project. While you will possibly have the same child on your hard drive in two different places they will be in two separate projects. This is useful in a custom equipment scenario when you want each project to be "self-supporting" or when you want maximum portability in your projects (e.g. burning them to a CD to send to a customer that uses Inventor)

Let's first look at a sample project path file (This tutorial is only dealing with how to modify your project file to use iParts. You should make yourself familiar with how project files work prior to making any modifications to your project file.)

[1] Autodesk Inventor R5.3 Online Help File

```
[Project Defaults]
MultiUser=FALSE
UseRelative=TRUE

[Included Path File]

[Workspace]
WorkSpace=.

[Local Search Paths]
Assembly=.\assembly
Details=.\details
AutoCAD2D=.\autocad2d
Parts=.\parts
Purchase=.\purchase
PDF=.\PDF

[Library Search Paths]
Iparts=i:\ipartfactories
_Iparts=.\purchase
```

Let's focus on the section of the section of the project file under the [Library Search Paths] section. The line `Iparts=I:\iparts` specifies the location of the iPart factories or parent files. In this case the iPart factories are being stored on a network drive mapped to the letter I and a folder named ipartfactories but you can named them whatever you like. You could also use the UNC path to the network drive if so desired (e.g. //NTServer/Inventor/ipartfactories). Under this folder you can further subdivide the parts into disciplines (e.g. a folder for hardware, fasteners, electrical, pneumatics etc. see Figure 1) but some thought must be put into this file structure.

IMPORTANT!

Since iParts are Library parts there is great danger in changing the names, locations and relative folder positions of the files. A bug in R5.3 will cause a change of the library folder to cause a near endless file resolve problem if you change the name or position of one of these folders. Library files should, once placed in any assembly, be left alone and in position (there are slight exceptions). Put a lot of thought into how you want to structure this folder before implementing it.

The location of the factories or parent parts is the same in each scenario (project specific vs. global). Typically you would want to place these factories on a drive where all users could access them (e.g. network drive) even if users are working on projects on their local drives.

The difference in the scenarios comes into play in the next line of the project file. The line `_Iparts=I:\iparts` specifies where the children will be created. This is known as a proxy path and is indicated by the underscore _ character preceding the word iParts. The names of the iPart factory and the proxy path must be the same. The underscore simply indicates that this is a proxy path indicating the iParts' children location.

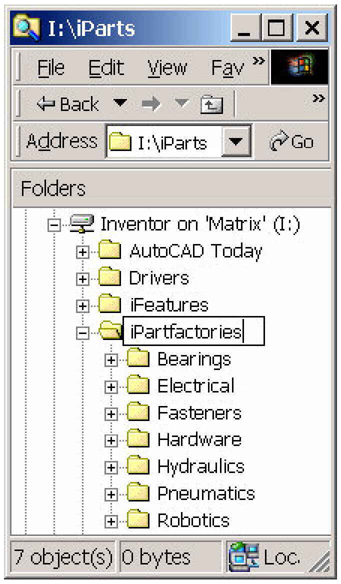

Figure 1 - Sample Ipart Factory File Structure

Let's look at an example proxy path for each scenario:

Global:

Since in this scenario we want the children to be available to all users on the network we want to place them on a network drive available to the users. Since they all have access to the parent or factory part location, this is an excellent candidate. A portion of a project file for this type of scenario might look like:

```
[Library Search Paths]
Iparts=I:\ipartfactories
_Iparts=I:\ipartchildren
```

In this example the children of the iparts will be stored under the folder named `I:\ipartchildren` located on the "I" drive (again you can use UNC file paths). All projects and all users can now reference this proxy path location. If User A creates a child part (say named ABC.ipt) then when User B chooses to place this same child when insert an iPart factory then the existing child part (ABC.ipt) will be used. This ensures that all users are using the same iParts and hence the same filenames, descriptions, part numbers and other properties that are stored in the iParts.

If no proxy path is defined the iPart children will be created in a folder underneath the location of the iPart factory. While some users like having both the children and factories in one place I find that it is easier to handle them into two separate directories.

Again, the advantage to this method is that you do not have extra, duplicate, iPart children on your system. Also since all ipart children are in one folder it is easy to back up these parts.

You can simplify the project file by adding a included project file in this scenario. An included project file is another project file that you can add to the current one. This is useful when you have the same information used in a lot of different projects. To do this make a new project file with a workspace of . (a period which is the relative path marker). Add only your library paths to this file. The project fill will look like:

```
[Project Defaults]

[Included Path File]

[Workspace]

[Local Search Paths]

[Library Search Paths]
Iparts=i:\ipartfactories
_Iparts=i:\ipartchildren
```

Save this file in a location available to every user. Now edit your main project file to remove the two library search paths and to add the included project file. Your project file will look like:

```
[Project Defaults]
MultiUser=FALSE
UseRelative=TRUE

[Included Path File]
Included Pathfile=H:\Projects\Project_Path_Files\Library Paths.ipj

[Workspace]
WorkSpace=.

[Local Search Paths]
Assembly=.\assembly
Details=.\details
AutoCAD2D=.\autocad2d
Parts=.\parts
Purchase=.\purchase
PDF=.\PDF

[Library Search Paths]
```

So the information in the included path file will be added to the information in this project file. In the projected browser in Inventor you will see that the paths have been added. Keep in mind that you cannot have any relative paths addressed in the included file as it is usually stored in a non project folder. As above the paths needs to be hard coded so that they point to the correct directories. Furthermore since the location of the children changes in the Project Specific scenario (see below) an included file can only contain the iPart Factory location. For the Project Specific scenario I suggest not using included path files.

Project Specific:

In this scenario we want iPart children to be created in the scope of the individual project. In this case the sample project file might look like:

```
[Library Search Paths]
Iparts=I:\ipartfactories
_Iparts=.\purchase
```

In this case the child parts will be created in the directory named `purchase` which is located in the root of the workspace (since we are using relative path names). The ipart children will be created in the purchase directory of this project (and each project if oyu keep the project files consistent between projects). This allows the users to pack & go the entire project (or since all parts are located under one master folder in this example, just copy the folder) onto another drive or CD. When opening the assembly you will be asked to resolve the links to the iPart factory. You can simply skip these resolutions and the assembly will open with no problems. Only if you wanted to change the instance of one of the iParts would you run into problems. However in a situation where you are archiving projects onto CD (or sending a CD to a customer for review) this would not be an issue. If you did need to change the instance of an iPart simply copy the CD onto your drive and with the proper project file the iPart children will be reconnected to their parents.

While the basic location of the storage of the ipart children is controlled by the proxy path there are additional folders that are created when placing iParts. When an iPart is placed in the assembly the child will be created in a directory:

`.\purchase\X\Y\Z.ipt` (project specific) or
`I:\ipartchildren\X\Y\Z.ipt` (global)

Where X = the subfolder of the iPart factory location (e.g. Pneumatics, Hydraulics etc. as shown above), Y = a folder with a name the same as the name of the iPart factory part and Z is the filename specified in the specific iPart table.

For example if we had an iPart named Washer.ipt located in:

`I:\ipartfactories\Fasteners\Washer.ipt`

And the filename defined in the iPart table for the child that we are placing was `Washer-0.25.ipt` then the child would be created as:

`.\purchase\Fasteners\Washer\Washer-0.25.ipt` (project specific) or
`I:\ipartchildren\Fasteners\Washer\Washer-0.25.ipt` (global)

These folders are created on the first placement of a specific iPart. Any other children of `Washer.ipt` would be placed in this directory as well.

Now that you have created your project path file for dealing with iParts it's time to start building your library. A great source for iParts is at Charles Bliss' site:

http://www.cbliss.com/inventor/

Charles has a wealth of parts available for download. He also will host the iParts you have created so that others can use them as well. If you want to learn how to create your own iParts I suggest reading my three part tutorial on this subject located at my site:

http://www.sdotson.com

Once you have downloaded a part from his site the first thing to do is to unzip it into the correct directory. Put the .ipt in the correct directory as specified in your project path file. Be sure that it is in the correct place as moving it later can cause problems with your assembly file resolution.

Open the part in Inventor (sinceR6 will not allow you to edit library parts, I suggest you make a project specifically for editing iParts. It should have the location of your factory folder (e.g. I:\ipartfactories) as the workspace and nothing else. This will allow you to edit the files.) The first thing you will notice is that there is likely nothing on the screen. Look at the browser. You will probably see that the End Of Part (EOP) marker is rolled up tot he top. This compresses the file for ease of downloading. Move the EOP marker to the bottom of the browser window. (see Figure 2)

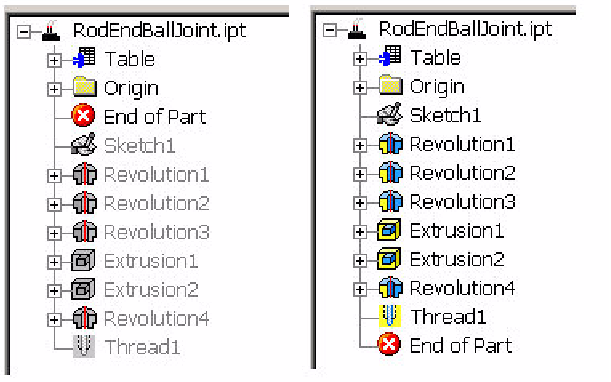

Figure 2 - EOP Marker Rolled Up & Down

Now save the file. Likely you will be asked if you want to save the files as a current version of Inventor. This is due to the fact that many of these parts were created in previous versions of Inventor. Click OK and save the part. You are now ready to insert this part into an assembly.

A final note about Library Paths; Library paths are not only used for iParts but can be used for standard parts as well. These are standard parts that will not be edited (like motors, bearing etc..) to which you want all users to have access. These can be added directly to your project file or to your master included path file. Just keep in mind that the same restrictions of not renaming or moving files applies to ALL library files (iParts and normal parts). Inventor does allow you to add Library paths "on the fly" without closing all files (unlike all other paths). Simpy RMB on the Library Path Section and choose Add to add a path and have instant access to that folder.

In many situations defining these collections of parts as libraries is not necessary. You could just as easily define them as Workgroup or Local Search Paths. I would suggest reading up on project files before defining any library paths.

I hope this has helped to explain how iParts are used and generated in Inventor. It's a good idea to do some extra reading on project files in general as this is the most understood and, in my opinion, most important aspect to a successful project in Inventor.